WHAT IS CLAIMED IS:

1. A system for performing register renaming of source registers in a processor having an instruction window for storing a group of instructions to be executed by the processor, wherein new instructions are added to the instruction window when the processor retires preceding instructions, the system comprising:

first means for storing source and destination register addresses for the instructions in the instruction window;

second means, coupled to said first means, for accessing said stored source and destination register addresses for performing a data dependency check for each new instruction added to the instruction window; and

third means, coupled to said second means, for renaming source register addresses for instructions having dependencies as determined by said second means.

2. The system of claim 1, wherein the system further comprises a rename result register file for storing said renamed source register addresses.

3. The system of claim 1, wherein the instruction window is a variable advance instruction window.

4. The system of claim 3, wherein instructions in the variable advance instruction window are assigned a tag, and the tag of an instruction leaving the window is assigned to the next new instruction to be added to the variable advance instruction window.

5. The system of claim 1, wherein said dependencies are input dependencies.

6. The system of claim 1, wherein:

said second means determines whether more than one dependency exists; and

a priority encoder, coupled to said second and third means, which selects a highest priority dependency identified by said second means and passes said highest priority dependency to said third means.

7. The system of claim 6, wherein said system further comprises a temp buffer means for storing results of instructions executed by the processor according to said tags to avoid output and anti-dependencies, wherein said ternp buffer permits the processor to execute instructions out of order and in parallel.

8. The system of claim 7, wherein said third means comprises tag assignment logic for determining where in said temp buffer operands of dependent instructions are located according to said highest priority dependency.

9. The system of claim 8, wherein said system further comprises means for passing the results stored in said temp buffer to a main register file in program order.

10. A method for performing register renaming of source registers in a processor having a variable advance instruction window for storing a group of instructions to be executed by the processor, wherein new instructions are added to the variable advance instruction window when a location becomes available therein, the method comprising the steps of:

(a) storing source and destination register addresses for the instructions in the variable advance instruction window;

(b) assigning a tag to each instruction in the variable advance instruction window, wherein the tag of each retired instruction is assigned to the next new instruction to be added to the variable advance instruction window;

(c) storing, in a temp buffer, results of instructions executed by the processor according to their corresponding tags to avoid output and anti-dependencies, said temp buffer permitting the processor to execute instructions out of order and in parallel;

(d) performing data dependency checks for input dependencies for each new instruction added to the variable advance instruction window;

(e) determining where operands are located in the temp buffer for the instructions having input dependencies as determined by step (d);

(f) renaming source register addresses of the instructions having dependencies; and

(g) storing said renamed source register addresses in a rename result register file.

11. The method of claim 10, further comprises the step of passing the results stored in the ternp buffer to a main register file in program order.